

1/18

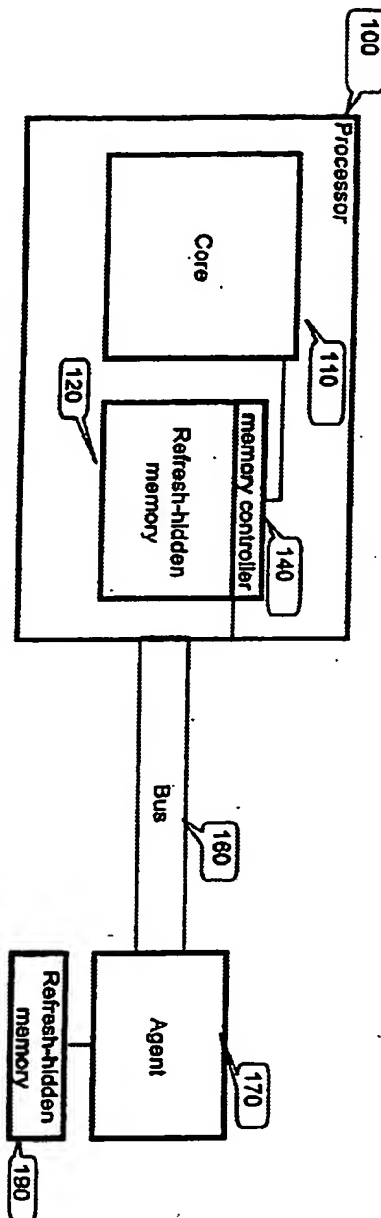


Figure 1

BEST AVAILABLE COPY

2/18

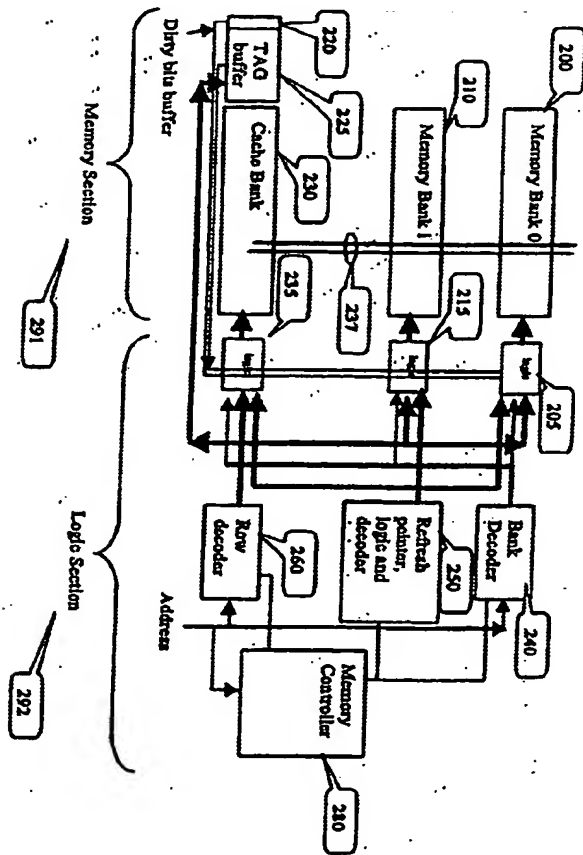
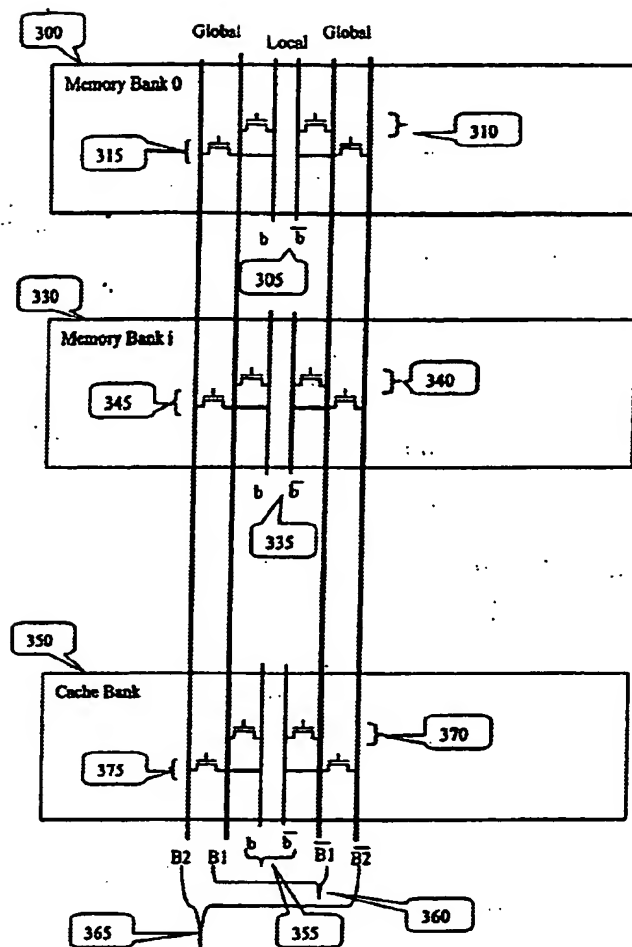


Figure 2

BEST AVAILABLE COPY

3/18

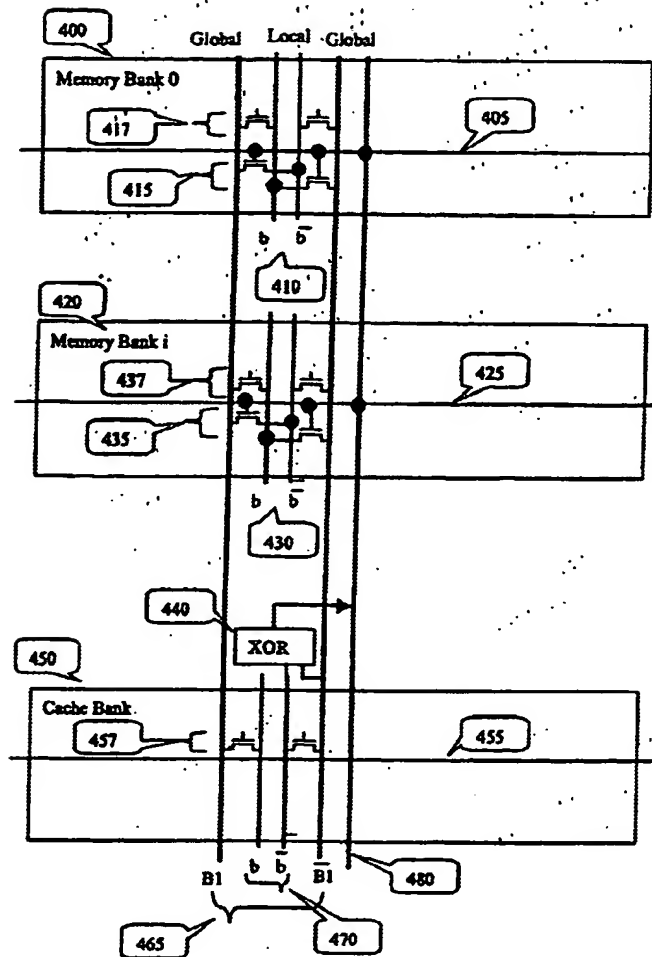
Figure 3



BEST AVAILABLE COPY

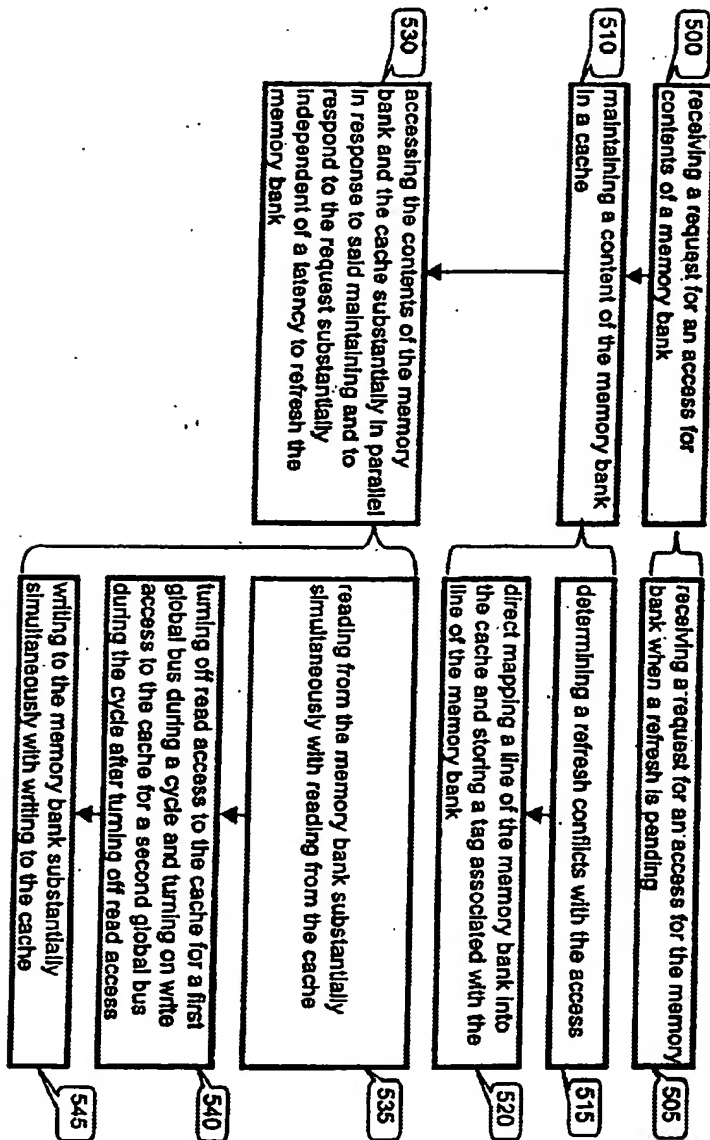
4/18

Figure 4



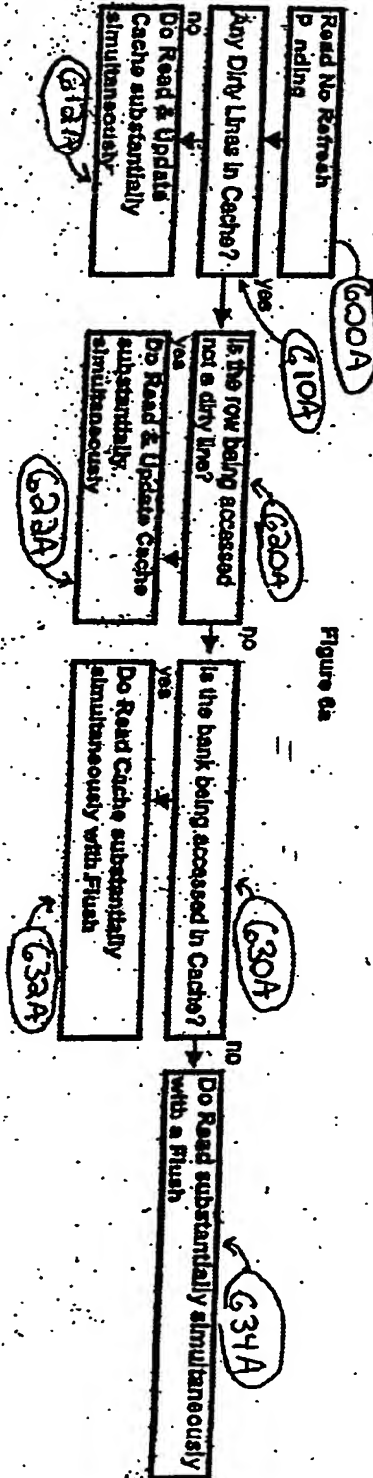
BEST AVAILABLE COPY

5/18



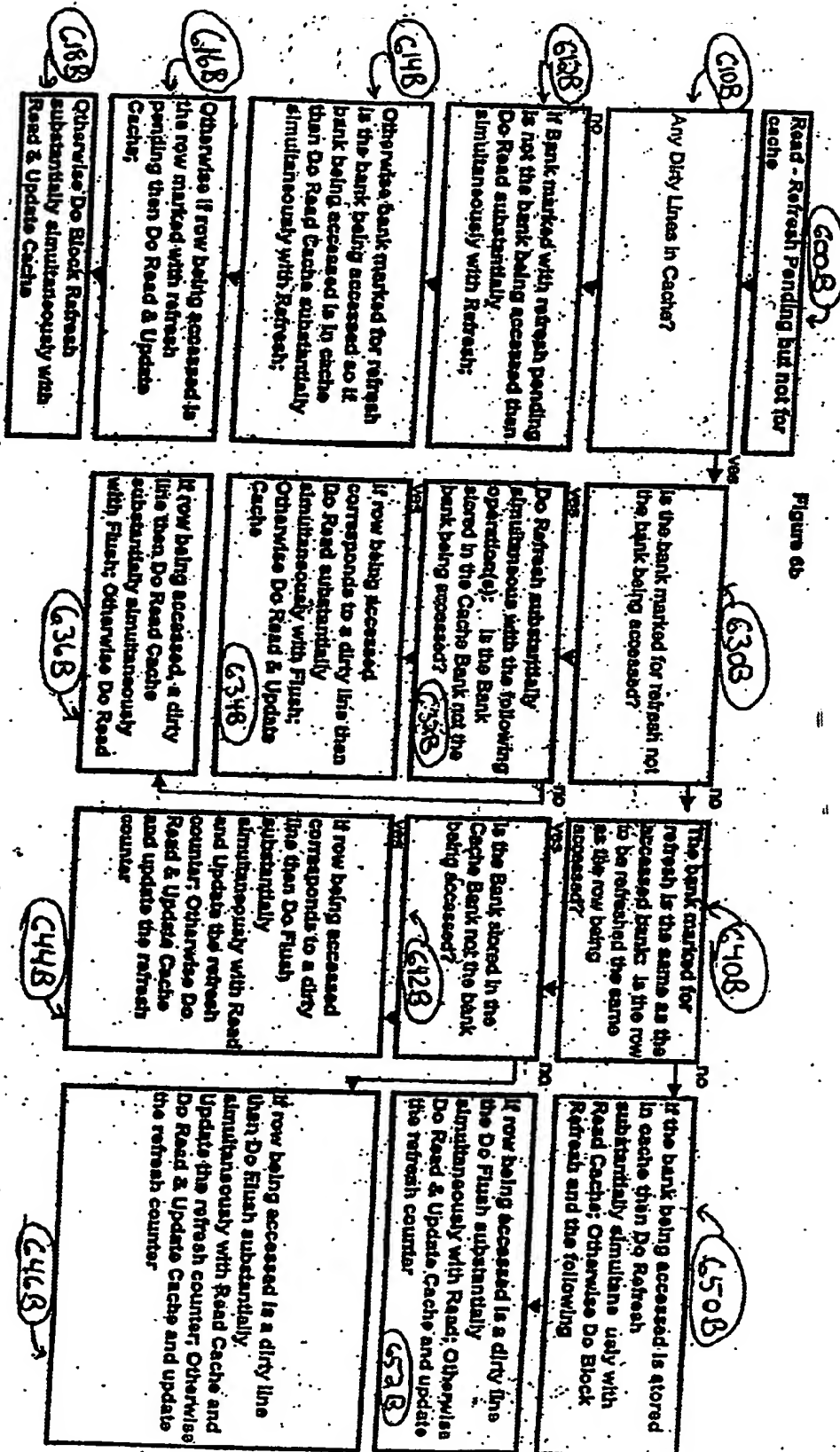
BEST AVAILABLE COPY

6/18

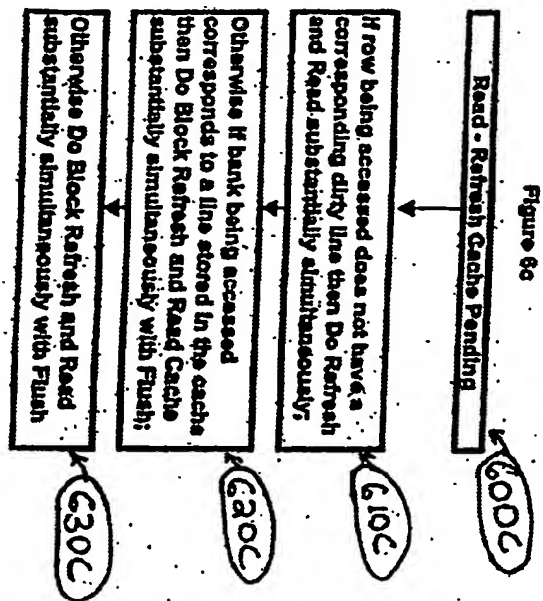


Legend for Figures 6a, b, &c  
 Read = read the row being accessed in the memory bank  
 Read Cache = read the cache version of the row being accessed  
 Update Cache = write the row from the memory bank being accessed into the cache and substantially simultaneously store a cache tag for the entry  
 Refresh = refresh the row marked as refresh pending  
 Block Refresh = prevent the refresh of the row until an action may be completed

7/18

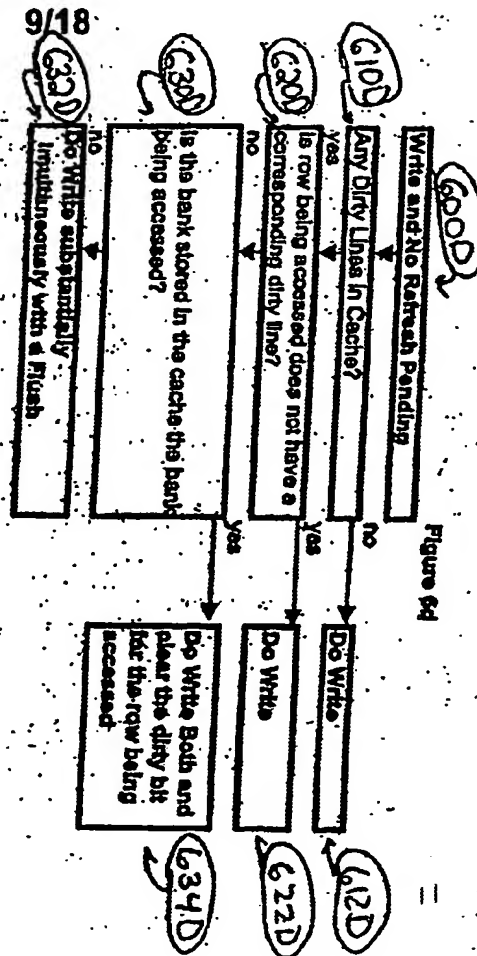


8/18



BEST AVAILABLE COPY

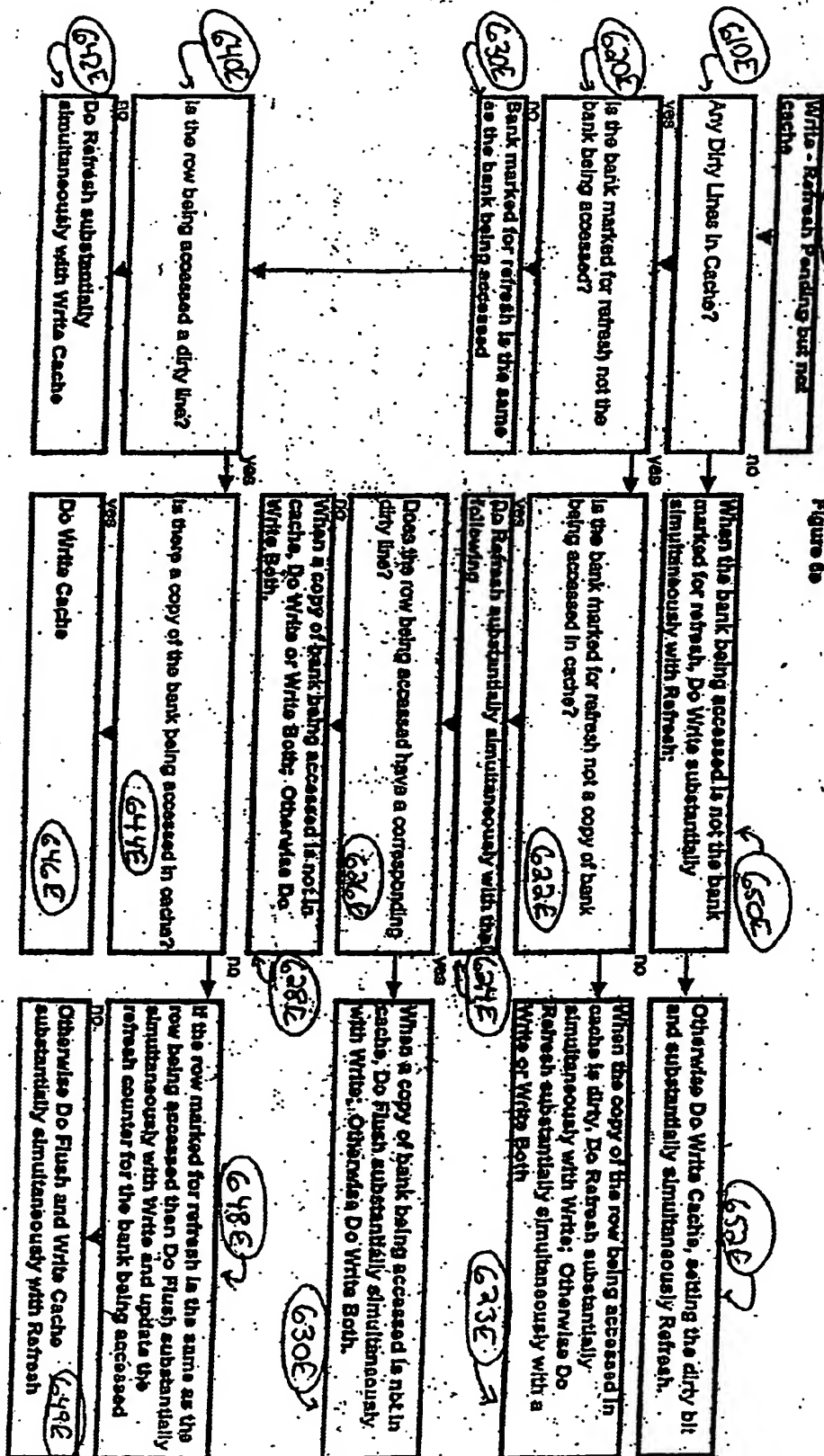




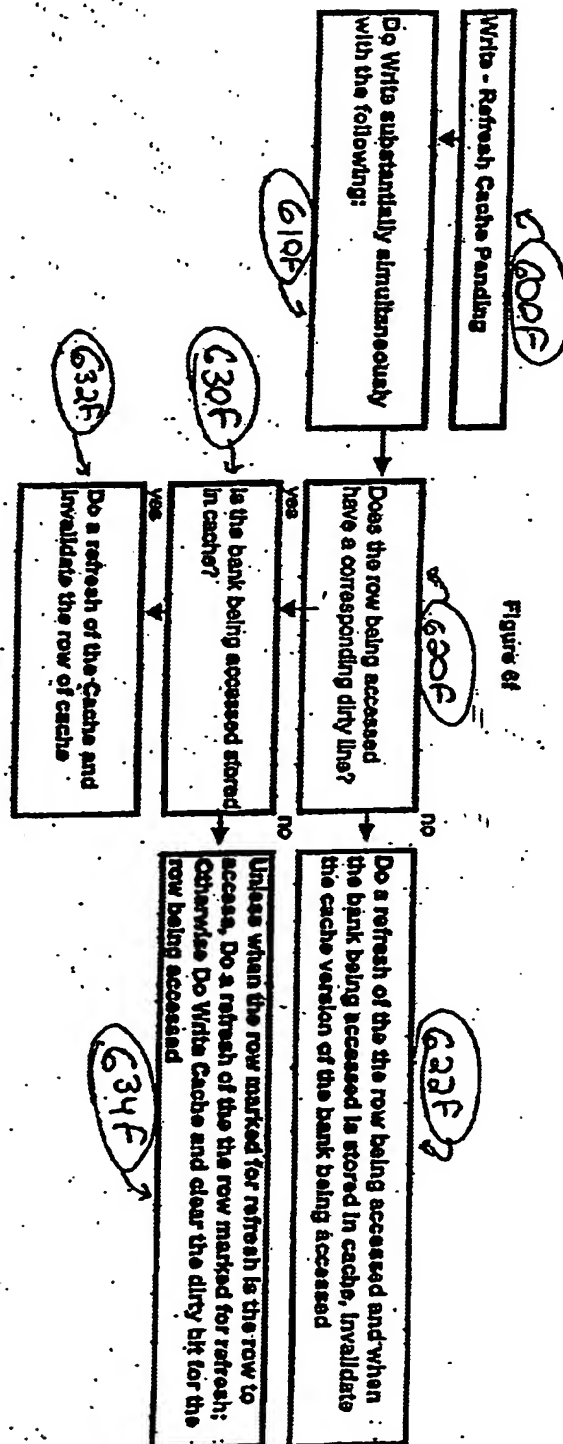
Legend for Figures 6d, e, & f  
 Write = write to the row being accessed in the memory bank  
 Write Cache = Write to the cache version of the row being accessed  
 Update Cache = write the row from the memory bank being accessed into the cache and substantially simultaneously store a cache tag for the entry  
 Refresh = copy the cache version of a row into the corresponding bank substantially simultaneously with clearing the dirty bit for the row  
 Block Refresh = prevent the refresh of the row until an action may be completed

BEST AVAILABLE COPY

10/18



11/18



BEST AVAILABLE COPY

12/18

Figure 7A

## Inputs

Refresh Counter: rb, rr /\* refresh bank, refresh row \*/  
 Current Access Address: wb, wr /\* access (write) bank and row \*/  
 Number of rows in a bank: row

## Buffers

Valid[row]  
 Dirty[row]  
 Cache[row]  
 Cachetag[row] /\* valid bits for each of the rows in the cache bank \*/  
 /\* dirty bits for cache lines \*/  
 /\* cache bank \*/  
 /\* Tag for cache lines \*/

## Functions

RefreshO  
 FlushO  
 ReadUpdateCO  
 ReadO  
 WriteO  
 ReadCacheO  
 WriteCacheO  
 WriteBothO  
 {mem[rb, n] = mem[rb, rr];}  
 {mem[cachetag[wr], wr] = cache[wr] \$\$ dirty[wr] = 0;} /\* WB dirty line \*/  
 {data\_out = mem[wb, wr] \$\$ cache[wr] = mem[wb, wr]  
 \$\$ cachetag[wr] = wb; }  
 {data\_out = mem[wb, wr];}  
 {mem[wb, wr] = data\_in; }  
 {data\_out = cache[wr] \$\$ cache[wr] = cache[wr];}  
 {cachetag[wr] = data\_in \$\$ cachetag[wr] = wb \$\$ dirty[wr] = 1;}  
 {WriteO \$\$ WriteCacheO \$\$ dirty[wr] = 0; }  
 /\* write to both the cache and memory \*/

(\$\$ - means operations are done in parallel)

## pseudo program description

/\* Initialize all cachetags to be NULL \*/  
 Cachetag[\*] = NULL;

\*\*\* READ \*\*\*

/\*----- No refresh pending ----- \*/  
 if (no refresh pending) {  
 if (no dirty line) {ReadUpdateCO; }  
 else /\* there is a dirty line \*/ {  
 if (!dirty[wr]) { ReadUpdateCO; }  
 else if (wb == cachetag[wr]) { ReadCacheO \$\$ FlushO \$\$ dirty[wr] = 0; }  
 else {

BEST AVAILABLE COPY

13/18

Figure 7B

```

/* -----Refresh pending -----*/
else {
  If (rb != cache) {
    If (no dirty line) {
      if (rb != wb) { Read0 $$ Refresh0; }
      else { /* rb == wb */
        if (wb == tag[wr])
          { ReadCached0 $$ Refresh0; }
        else {
          if (wr == r) ReadUpdateC0;
          else Block_refresh $$ ReadUpdateC0;
        }
      }
    }
    else /* there is a dirty line */
      if (rb != wb) {
        Refresh0 $$
        if (tag[wr] != wb) {
          if (dirty[wr]) { Flush0 $$ Read0; }
          else ReadUpdateC0;
        }
        else {
          if (dirty[wr]) { ReadCached0 $$ Flush0; }
          else Read0;
        }
      }
    /* end if rb != wb */
    else { /* refresh bank is the same as the accessed bank */
      if (wr == r) {
        if (tag[wr] != wb) {
          if (dirty[wr]) { Flush0 $$ Read0
            $$ UpdateRefreshCounter0; }
          else { ReadUpdateC0 $$ Update Refresh Cnt0; }
        }
        else {
          if (dirty[wr]) { ReadCached0 $$ Flush0
            $$ UpdateRefreshCnt; }
          else { ReadUpdateC0 $$ UpdateRefreshCounter0; }
        }
      }
    }
    if (tag[wr] == wb) { Refresh0 $$ ReadCached0; }
    /* dirty status for the line unchanged */
    else {
      BlockRefresh;
      if (dirty[wr]) { Flush0 $$ Read0; }
      else ReadUpdateC0;
    }
  }
}

```

BEST AVAILABLE COPY

14/18

```
else /* the cache bank is being refreshed */ {  
    if (dirty[wr]) {  
        if (tag[wr] == vb) {  
            BlockRefresh;  
            ReadCacheQ $$ FlushQ;  
        }  
        else {  
            BlockRefresh;  
            ReadQ $$ FlushQ;  
        }  
    }  
    else {  
        RefreshQ  
        ReadQ  
    }  
} /* end refresh pending */
```

Figure 7C

BEST AVAILABLE COPY

Figure 7D

```
/** WRITE **/  
/* ----- No pending refresh ----- */  
if (no refresh pending) {  
    /* there is no dirty line */  
    if (no dirty line) { WriteO; }  
    else { /* there is a dirty line */  
        if (!dirty[wr]) { WriteO; }  
        else {  
            if (Cachetag[wr] == wb) { WriteBothO $$ dirty[wr] = 0; }  
            else { WriteO $$ FlushO; }  
        }  
    } /* end dirty line */  
}
```

16/18

Figure 7E

```

/* ----- Pending refresh ----- */
else {
    if (rb != cache) {
        if (no dirty line) {
            if (wb != rb) { Write0 $$ Refresh0 }
            else { WriteCache0 $$ dirty[wr] = 1 $$ Refresh0 ; }
        }
        else { /* there is a dirty line */
            if (rb != wb) { /* refresh bank is different from the access bank */
                if (rb != cachetag[wr]) {
                    Refresh0 $$
                    If (dirty[wr]) {
                        If (cachetag[wr] != wb) { Flush0 $$ Write0 ; }
                        Else { Writeboth0 ; }
                    }
                    else {
                        If (cachetag[wr] != wb) { Write0 ; } /* or writeboth0 */
                        Else { Writeboth0 ; }
                    }
                }
                else {
                    if (dirty[wr]) { Refresh0 $$ Write0 ; } /* cache line remains dirty */
                    else { Refresh0 $$ Write0 ; } /* Or writeboth0 */
                }
            }
            else { /* refresh bank is the same as the access bank */
                If (dirty[wr]) {
                    If (cachetag[wr] == wb) { Writecache0 ; }
                    else {
                        if (wr == r) { /* lucky day */
                            Flush0 $$ write0 $$ Updaterefreshcounter;
                        }
                        else {
                            Flush0 and writes to cache with new data ;
                            $$ Refresh0;
                        }
                    }
                }
            }
        }
        else { /* the line is not dirty */
            /* create another dirty line */
            Refresh $$
            WriteCache0;
        }
    }
}

```

BEST AVAILABLE COPY



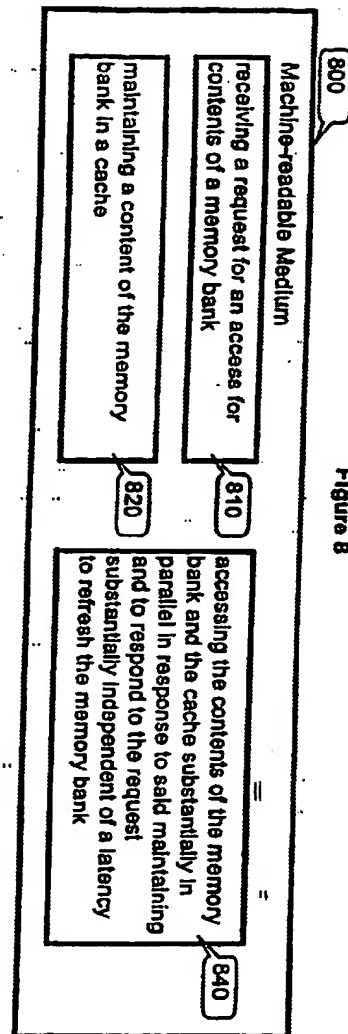
17/18

```
else { /* cache bank is being refreshed */  
  Write0 $$  
  if (dirty[wr]) {  
    if (cachetag[wr] == wb) {  
      Refresh0 $$  
      Cachetag[wr] = NULL;  
    }  
    else {  
      Refresh0;  
      /* unless wr = rr then we writeback and clear dirty bit */  
    }  
  }  
  else { /* not dirty */  
    Refresh0;  
    if (cachetag[wr] == wb) { Cachetag[wr] = NULL; }  
  }  
}
```

Figure 7F

BEST AVAILABLE COPY

18/18



BEST AVAILABLE COPY